



TITLE:

正則な論理関数列を実現する論理回路の複雑さ(計算機構に関する数学的基礎理論とその応用)

AUTHOR(S):

安浦, 寛人; 矢島, 脩三

CITATION:

安浦, 寛人 ...[et al]. 正則な論理関数列を実現する論理回路の複雑さ(計算機構に関する数学的基礎理論とその応用). 数理解析研究所講究録 1983, 494: 248-258

ISSUE DATE:

1983-06

URL:

<http://hdl.handle.net/2433/103560>

RIGHT:

正則な論理関数列を実現する論理回路の複雑さ

京都大 工 安浦 寛人 (Hiroto YASUURA)

矢島 脩三 (Shuzo YAJIMA)

1. まえがき

集積回路技術の進歩により、大規模な論理回路の構成が可能となり、種々の演算・操作を実現する大規模論理回路が設計されはじめている。このような、大規模な論理回路の設計に対しては、従来の論理回路設計法では、性能の良い回路の設計が難しく、新しい大規模論理回路に適した設計手法が必要となる。我々は、問題の性質を利用して、効率の良い論理回路を設計する手法として、論理関数列に基づく方法を提案している(1)。本稿では、最も基本的で、実用的にも重要な正則な論理関数列について、それを実現する論理回路の複雑さについて議論する。

正則な論理関数列は、形式言語理論における正則集合に対応し、論理和、論理積、パリティ関数等の基本的関数や、比較器、モジュロ計数器、加算器等の設計とも密接な関連を持つ。本稿においては、入次数制限のある一定遅延を持つ論理素子を用いて構成された論理回路と、入出力の境界配置条件(2)を満たすVLSIモデルを考え、素子数、計算時間、VLSIの面積等を評価尺度として正則な論理関数列を実現

する論理回路の複雑さの評価を行う。

2. 論理回路とVLSIモデル

ここでは、入次数制限のある一定遅延を持つ論理素子を用いて構成される論理回路を考える。簡単のため入次数は、2以下とする。

[定義 1] 論理回路は、各節点の入次数が2以下の有向グラフ $C=(V,E)$ である。

$V=V_C \cup V_I \cup V_O$ は節点の集合で、 V_C 、 V_I 、 V_O は、互いに交わらない部分集合である。 V_C 中の節点は、計算点と呼ばれ、 $\{f(x,y) \mid f:\{0,1\}^2 \rightarrow \{0,1\}\}$ 中の論理関数 f を実現する素子に対応する。各計算点は、1単位時間の遅延を持ち、その点への2つの入力枝の始点の論理値から f を計算する。 V_I 中の節点は、入次数が0で、入力点と呼ばれ、外部から論理値を与えることができる。 V_O 中の節点は、入次数が1かつ出次数が0で、出力点と呼ばれ、その論理値は入力枝の始点の論理値と、各時刻で一致する。 C 中の計算点の数を、論理回路 C の素子数と呼び、 $\text{size}(C)$ で表す。

[定義 2] 論理回路 C が、関数 $f:\{0,1\}^n \rightarrow \{0,1\}^m$ を時間 t で計算するとは、入力変数の集合 $I=\{x_1, x_2, \dots, x_n\}$ 、出力変数の集合 $O=\{y_1, y_2, \dots, y_m\}$ 、時刻の集合 $T=\{0, 1, \dots, t\}$ に対し、2つの写像 $h_1:I \rightarrow V_I \times T$ 、 $h_2:O \rightarrow V_O \times T$ があって、

$$(y_1, y_2, \dots, y_m) = f(x_1, x_2, \dots, x_n)$$

$$x_i = res(h_1(x_i)) \quad (i=1,2,\dots,n)$$

$$y_i = res(h_2(y_i)) \quad (i=1,2,\dots,m)$$

となることである。ここに、 $res(v,t)$ は、時刻 t における節点 v の論理値を表す。

h_1 、 h_2 が存在する t の最小値を関数 f の回路 C による計算時間と呼び、 $delay(C,f)$ で表す。

次に、VLSIモデルを定義する。

[定義 3] 論理回路 C のVLSIへの埋め込み E とは、次の(1)~(4)を満たす2次元平面へのグラフの埋め込みである。

(1) 回路 C は、平面上の凸領域 R の中に埋め込まれる。

(2) C 中の計算点、入力点、出力点はそれぞれ R 中の互いに重ならない $l \times l$ の正方形に割り当てられる。これらの正方形は、結線とも重ならない。

(3) C 中の枝 (v,v') に対し、 R 中で v と v' に対応する2つの正方形を連結する結線が存在する。

(4) 結線は l 以上の幅を持ち、 R 中の任意の点で重なり合うことのできる結線の数、ある定数以下である。

さらに、次の(5)を満たす時、その埋め込みは、境界配置条件を満たすという。

(5) C 中の入力点及び出力点に対応する正方形は、すべて R の境界に接する。

回路Cを埋め込みEによって、VLSIへ埋め込んだ時の面積を、 $\text{area}_E(C)$ と表し、CのEによる埋め込みの面積という。Cのあらゆる埋め込みを考えた時の面積の最小値をCの面積複雑度といい、 $\text{area}(C)$ と表す。

3. 正則な論理関数列

論理関数列 $\{f_n\}$ とは、各自然数 n に対し、ちょうど1つずつの n 変数論理関数 $f_n: \{0,1\}^n \rightarrow \{0,1\}$ を含むような論理関数の無限列(無限集合)である(1)。論理関数列 $\{f_n\}$ の生成言語 $L_{\{f_n\}}$ とは、 $\{0,1\}$ 上の系列の集合で、各自然数 n に対し、

$$\begin{cases} f_n(x_1, x_2, \dots, x_n) = 1 \rightarrow x_1 x_2 \dots x_n \in L_{\{f_n\}} \\ f_n(x_1, x_2, \dots, x_n) = 0 \rightarrow x_1 x_2 \dots x_n \notin L_{\{f_n\}} \end{cases}$$

(但し、 $x_i \in \{0,1\}$)となるものである。この対応関係により、 $\{0,1\}$ 上の系列の集合と論理関数列は1対1に対応することがわかる(1)。

[定義 4] 論理関数列 $\{f_n\}$ の生成言語 $L_{\{f_n\}}$ が正則集合(Regular Set)であるとき、 $\{f_n\}$ を正則な論理関数列と呼ぶ。

[例] $\{f_n \mid f_n(x_1, x_2, \dots, x_n) = x_1 x_2 \dots x_n\}$ は正則な論理関数列で、生成言語 $L_{\{f_n\}}$ は 11^* で表される正則集合である。

この他、論理和、パリティ関数、2つの2進数の大小比較等も正則な論理関数列であ

る。

[定義 5] $\{f_n^1\}$ 、 $\{f_n^2\}$ 、 \dots 、 $\{f_n^k\}$ (k は定数)を正則な論理関数列とする。このとき、

$\{f_n = (f_n^1, f_n^2, \dots, f_n^k) \mid f_n: \{0,1\}^n \rightarrow \{0,1\}^k\}$ を正則な関数列という。

[例] $\{f_n \mid f_n: \{0,1\}^n \rightarrow \{0,1\}^2, (f_n^1, f_n^2) = f_n(x_1, x_2, \dots, x_n) \text{ は } x_1 x_2 \dots x_n \text{ 中の } 1 \text{ の数の modulo}$

$3 \text{ の } 2 \text{ 進表現}\}$ は正則な関数列で、 $\{f_n^1\}$ の生成言語は $0^*10^*1(0^*10^*10^*1)^*0^*$ で表される

正則集合であり、 $\{f_n^2\}$ の生成言語は $0^*1(0^*10^*10^*1)^*0^*$ で表される正則集合である。

4. 正則な関数列を計算する回路の複雑さ

関数列 $\{f_n\}$ に対し、その中の関数 $f_n: \{0,1\}^n \rightarrow \{0,1\}^k$ を計算する回路 C_n の列を $\{C_n\}$ とすると、回路列 $\{C_n\}$ は関数列 $\{f_n\}$ を計算するという。正則な(論理)関数列 $\{f_n\}$ を計算する回路列 $\{C_n\}$ の複雑さに関し、次の定理が成立する。

[定理 1] $\{f_n\}$ を正則な(論理)関数列とする。任意の正整数 N に対し、 $n > N$ なる n が存在し、ある $i(1 \leq i \leq k)$ があって、 f_n^i が真に n 変数に依存するとき、

1) $\{f_n\}$ を計算する任意の回路列 $\{C_n\}$ に対し、 $\text{delay}(C_n, f_n) = \Omega(\log n)$ が成立する。また、 $\text{delay}(C_n, f_n) = O(\log n)$ となる $\{f_n\}$ を計算する回路列 $\{C_n\}$ を構成できる。

2) $\text{size}(C_n) = \theta(1)$ 、 $\text{area}(C_n) = \theta(1)$ なる $\{f_n\}$ を計算する回路列 $\{C_n\}$ が存在する。

3) $\{f_n\}$ を計算する任意の回路列 $\{C_n\}$ に対し、

$$\text{size}(C_n)(\text{delay}(C_n, f_n))^r = \Omega(n(\log n)^{r-1})$$

であり、この下界を実現する回路列 $\{C_n\}$ が構成できる。但し、 $r \geq 1$ である。

4) $\{f_n\}$ を計算する任意の回路列 $\{C_n\}$ に対し、

$$\text{area}(C_n)(\text{delay}(C_n, f_n))^r = \Omega(n(\log n)^{r-1})$$

であり、この下界を実現する回路列 $\{C_n\}$ が構成できる。但し、 $r \geq 1$ である。

5) 境界配置条件を満たす埋め込みの下で、 $\{f_n\}$ を計算する

$$\text{area}(C_n)(\text{delay}(C_n, f_n))^r = O(n(\log n)^r)$$

となる回路列 $\{C_n\}$ とその埋め込みが構成できる。但し、 $r \geq 1$ である。

[証明] 1) 下界となることは、真に n 変数に依存する関数列の性質より明らか(3)。実際の回路構成法は文献(4)(5)を参照されたい。

2) 順序回路による $\{f_n\}$ の実現を考えればよい。

3) 真に n 変数に依存する論理関数 f を計算する回路 C について、

$$\text{size}(C)(\text{delay}(C, f))^r = \Omega(n(\log n)^{r-1})$$

となることは、次のようにして示せる。

(a) $\text{size}(C) \geq n/\log_2 n$ のとき

任意の回路において、真に n 変数に依存する f の計算時間は、 $\lceil \log_2 n \rceil$ 以上であるから、

$$\text{size}(C)(\text{delay}(C, f))^r \geq (n/\log_2 n) \lceil \log_2 n \rceil^r$$

$$= n(\log n)^{r-1}.$$

(b) $\text{size}(C) < n/\log_2 n$ のとき

入力点の数は、高々 $2\text{size}(C)$ 以下であるから、入力にかかる時間は $\lceil n/2\text{size}(C) \rceil$ 以上である。よって、

$$\text{size}(C)(\text{delay}(C, f))^r > \text{size}(C) \lceil n/2\text{size}(C) \rceil^r$$

$$\geq (1/2^r)n(\log n)^{r-1}.$$

これが下限となることは、次のような回路列 $\{C_n\}$ を構成することにより、示される。

回路構成法

i) $L_{\{f_n\}}$ を受理する有限オートマトン $M = (\{0, 1\}, Q, g, q_0, F)$ (但し、 $\{0, 1\}$: 入力アルファベット、 Q : 状態の集合、 $g: Q \times \{0, 1\} \rightarrow Q$ (状態遷移関数)、 $q_0 \in Q$: 初期状態の集合) を考える。

ii) Q 上の自己写像の集合 $\{p \mid p: Q \rightarrow Q\}$ の中で、 $p_0(q_i) = g(0, q_i)$ 、 $p_1(q_i) = g(1, q_i)$ によって定義された p_0 、 p_1 から生成される半群 G を考える。

iii) G 中の p_i 、 p_j に対し、 $p_i * p_j$ を計算する I 回路、入力から p_0 、 p_1 を生成する IP 回路、 $p(q_0) \in F$ のとき 1、 $p(q_0) \notin F$ のとき 0 を出力する Z 回路を構成する。これらの回路は、 n に依存しない素子数で構成できる (4)(5)。

iv) 図 1 のような回路 C_n を構成する。入力点 v_1, v_2, \dots, v_s および出力点 u_1, u_2, \dots, u_k への入出力変数の割り当てを、

$$h_1(x_i) = (v_i \bmod s, \lceil i/s \rceil - 1)$$

$$h_2(y_i) = (u_i, d_1 + d_2(\lceil \log_2 s \rceil + (n/s)) + d_3)$$

(但し、 d_1 、 d_2 、 d_3 はそれぞれIP回路、I回路、Z回路の遅延時間)

とすれば、計算時間は、 $d_1 + d_2(\log_2 s + (n/s)) + d_3$ となる。 $\text{size}(C_n)$ が $O(s)$ となるこ

とは明らかである。よって、 $s = n/\log_2 n$ とすると、

$$\text{size}(C_n)(\text{delay}(C_n, f))^r = O(n(\log n)^{r-1})$$

となる。

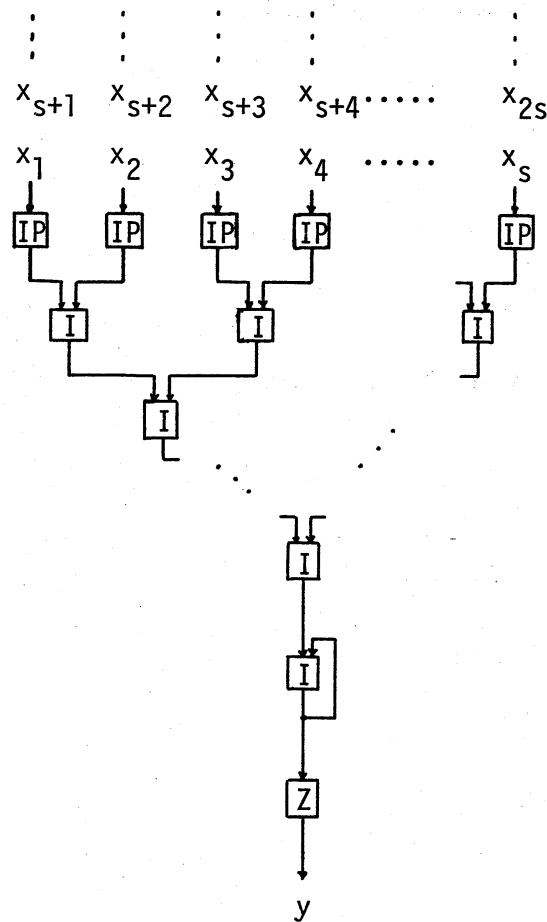


図 1. 正則な論理関数列を計算する回路 C_n

4)面積に関しては、3)とほぼ同様な議論で証明できる(6)。構成法には、3)と同じ回路列を用い、埋め込みはH木による。

5) 3)と同じ回路列 $\{C_n\}$ を考える。境界配置条件をみたす埋め込みにおいては、

$$\text{area}(C_n) = \Omega(n \log n)$$

となり、かつ

$$\text{area}_E(C_n) = O(n \log n)$$

なる埋め込みEは容易にわかる(2)。よって、

$$\text{area}(C_n)(\text{delay}(C_n, f_n))^r = O(n(\log n)^r)$$

となる。

証明終

定理1は、正則な(論理)関数列が、各種の複雑さの尺度に関して、最も簡単なクラスとなっていることを示している。実用上は、正則な(論理)関数列 $\{f_n\}$ に対し、 (f_1, f_2, \dots, f_n) (但し、 $f_i \in \{f_n\}$)を計算する問題がしばしば現れる。加算器の設計等は、このような問題の良い例である。

[定理 2] 正則な(論理)関数列 $\{f_n\}$ の部分列 (f_1, f_2, \dots, f_n) を計算する回路 C_n の列 $\{C_n\}$ の複雑さについて、

1) $\text{delay}(C_n, (f_1, f_2, \dots, f_n)) = O(\log n)$ となる回路列が存在する。

2) $\text{size}(C_n) = \theta(1)$, $\text{area}(C_n) = \theta(1)$ となる回路列が存在する。

3) $\text{size}(C_n)(\text{delay}(C_n, (f_1, f_2, \dots, f_n)))^r = O(n(\log n)^r)$ となる回路列 $\{C_n\}$ で境界配置条件を満たす埋め込みによって、

$$\text{area}(C_n)(\text{delay}(C_n, (f_1, f_2, \dots, f_n)))^r = O(n(\log n)^r)$$

となるものが、構成できる。

[証明] 2)は、順序回路による実現を考えればよい。1)と3)は、図2のような回路を考える。図中、I回路、IP回路、Z回路は、定理1の証明と同じであり、S回路は、状態 q と写像 p を入力して、状態 $q' = p(q)$ を出力するものである。

証明終

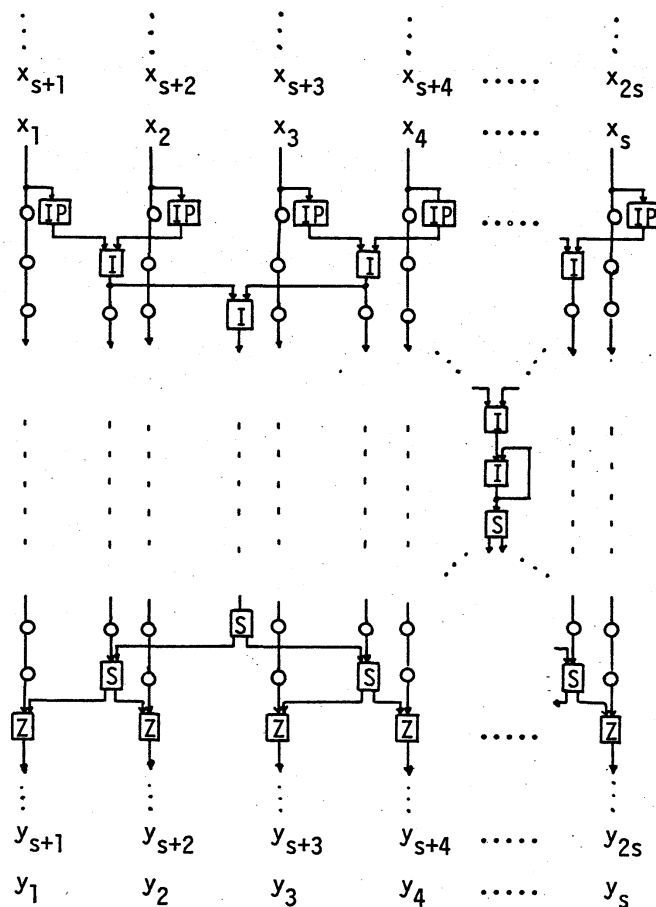


図 2 正則な論理関数列の部分列を計算する回路

謝辞 御討論頂いた本学上林弥彦助教授はじめ矢島研究室の諸氏に深謝いたします。本研究は、一部、文部省科学研究費補助金による。

参考文献

- (1)安浦寛人: 論理関数の複雑さの理論とその高速論理回路の構成法への応用、情報処理学会論文誌、21巻4号、pp.268-278、昭和55年7月。
- (2)安浦、矢島: 論理回路のVLSI上での面積について、電子通信学会論文誌、Vol. J65-D No.8、pp.1080-1087、昭和57年8月。
- (3) J. E. Savage: The Complexity of Computing, Jhon Wiley & Sons, 1976.
- (4)安浦、矢島: 論理関数を実現するのに必要な論理回路の段数について、電子通信学会論文誌、Vol. 62-D No.9、pp.561-568、昭和54年9月。
- (5) S. H. Unger: Tree Realizations of Iterative Circuits, IEEE Transactions on Computers, Vol.C26 No.4, pp.365-383, April 1977.
- (6)和田、萩原、都倉: VLSIモデルにおける面積複雑度、電子通信学会論文誌、Vol.J65-D No.4、pp.478-485、昭和57年4月。